总体设计说明书

网上书城系统

目录

1．引言 3
1.1 编写目的 3
1.2 项目背景 3
1.3 参考资料 3
2．任务概述 4
2.1 目标 4
2.2 开发环境 5
2.3 运行环境 7
2.4 需求概述 8
3．总体设计 10
3.1 总体结构设计 10
3.2 处理流程 11
4．数据库设计 15
4.1 数据库的分析 15
4.2 数据库的设计 15
4.2.1 数据库的设计结构 *15*
4.2.2 数据表的结构 *16*
4.2.2.1 商店图书表 16
4.2.2.2 商店图书类别表 16
4.2.2.3 商店管理员表 17
4.2.2.4 购书用户表 17
4.2.2.5 购物车订单号表 18
4.2.2.6 购物车订单货物列表 18
4.2.3 数据库的生成 *18*

1. 引言

1.1 编写目的

该项目的目的是开发一个电子书店网站以及后台管理系统。

1.2 项目背景

相信大家一定都有逛书店的经验，一般来说，逛书店的人可以分为，一种是根本不买书，只是想要看看吸收点咨询；另一种是要买书，而会对于书的进行比较，甚至是货比三家。对于纯粹逛书店的人来讲，常常是站到脚酸；同样对于需要买书的人而言，往往也要逛好几家书店，仔细翻阅内容才能购买。

所以，电子书店的出现，也随着网络的发达如雨后春笋般盛行起来，同样提供内容简介和封面图片，消费者一样可以在家里就拥有类似逛书店般的乐趣。

加上在搜索引擎这种咨询科技的帮助下，消费者很快就可以搜寻到想要的书籍；例如留言板、讨论区、E-MAIL 服务，更是为传统书店提供多元化的服务。

1.3 参考资料

[1] Bruce Eckel 著, 侯捷 译. Thinking in Java Second Edition. 机械工业出版社, 2002.

[2] 孙鑫. Java Web 开发详解－XML+XSLT+Servlet+JSP 深入剖析

与实例应用. 电子工业出版社, 2006.

[3] Roger S.Pressman. Software Engineering A Practitioner’s Approach Fifth Edition. 机械工业出版社, 2006.

[4] Paul DuBois. MySQL :The definitive guide to using, programming, and administering MySQL 4, Second Edition. Sams, 2004.

[5] Michael Kofler. The Definitive Guide to MySQL 5, Third Edition. Apress, 2006.

[6] Robin Dewson 著, 董明 译. SQL Server 2005 基础教程. 人民邮电出版社, 2006.

[7] Joseph Schmuller 著, 李虎 赵龙刚 译. UML 基础、案例与应用. 人民邮电出版社, 2006.

2. 任务概述

2.1 目标

用户注册；

用户登录；

查看用户的订单信息；

修改用户个人信息；

图书选购,(可按分类方式查找图书，或通过关键字进行查询)；

查看图书详细情况；

购物车功能；

后台管理部份:

商店图书管理：添加，修改，删除，查看；

用户信息管理：查看，修改，删除；

订单信息管理：查看订单清单，更新订单付款，出货状态，删除订单。

2.2 开发环境

网页设计：

Dreamweaver 8

J2EE 环境：

JCreator Pro 3.50

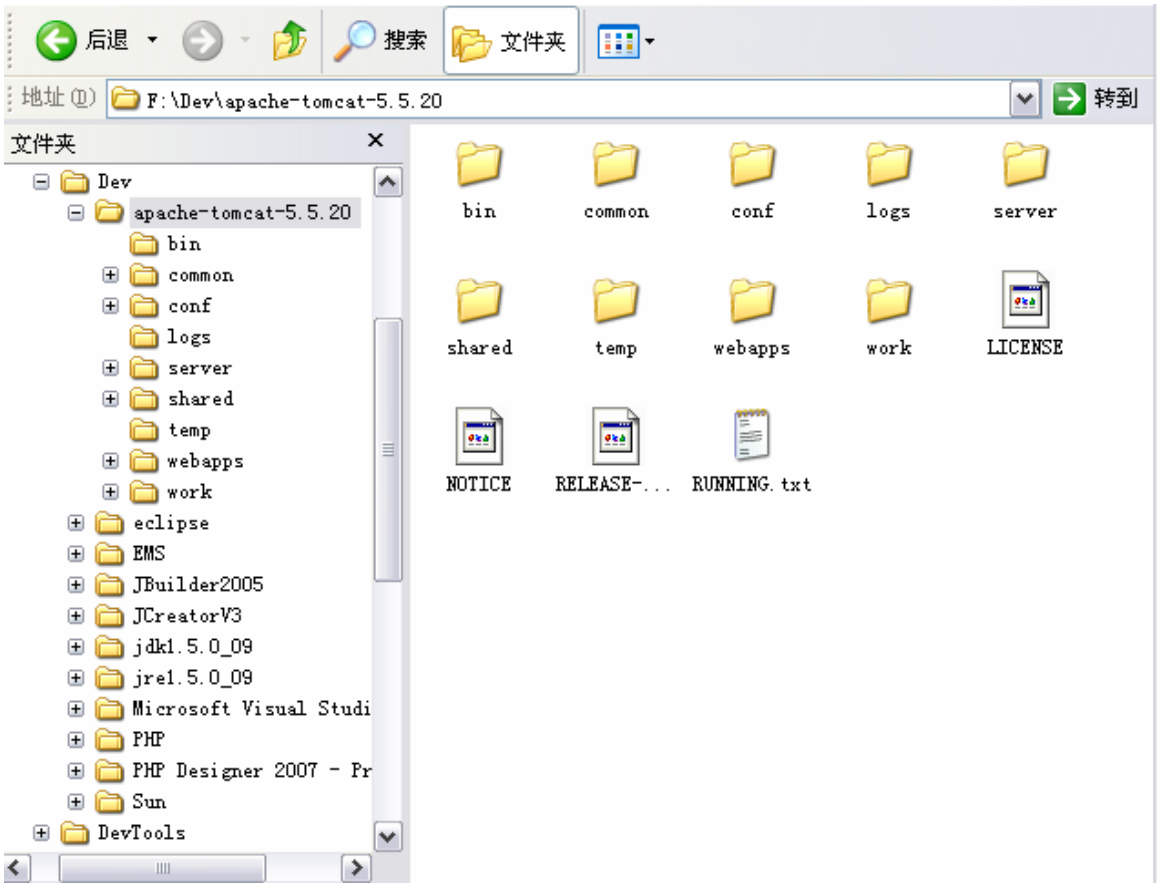
Eclipse SDK Version: 3.2.1

MyEclipse5.0 GA

Apache Tomcat 5.5.20

- 安装 Tomcat：

使用 WinRAR 等解压工具将 apache-tomcat-5.5.20.zip 解压到指定的驱动器和目录中。安装后的目录层次结构如图：


后退
搜索
文件夹
地址(D)
F:\Dev\apache-tomcat-5.5.20
转到
文件夹
Dev
apache-tomcat-5.5.20
bin
common
conf
logs
server
shared
temp
webapps
work
eclipse
EMS
JBuilder2005
JCreatorV3
jdk1.5.0_09
jre1.5.0_09
Microsoft Visual Studi
PHP
PHP Designer 2007 - Pr
Sun
DevTools
bin
common
conf
logs
server
shared
temp
webapps
work
LICENSE
NOTICE
RELEASE-...
RUNNING.txt

Tomcat 目录结构及用途

目录	
/bin	存放启动和关闭 Tomcat 的脚本文件
/common/lib	存放 Tomcat 服务器及所有 Web 应用程序都可以访问的 JAR 文件
/conf	存放 Tomcat 服务器的各种配置文件，其中包括 web.xml
/logs	存放 Tomcat 的日志文件
/server/lib	存放 Tomcat 服务器运行所需的各种 JAR 文件
/server/webapps	存放 Tomcat 两个 Web 应用程序:admin, manage
Shared/lib	存放所有 Web 应用程序都可以访问的 JAR 文件
/temp	存放 Tomcat 运行时产生的临时文件
/webapps	当发布 Web 应用程序时，通常把 Web 应用程序的目录及文件放到这个目录下
/work	Tomcat 将 JSP 生成的 Servlet 源文件和字节码文件放到这个目录下

2.3 运行环境

- 采用 Microsoft SQL Server 作为数据存储数据库，或者采用 MySql；

● 采用 JSP/SERVLET/JavaBean 技术作为系统的解决方案，使系统具有更好的可扩展性；

● 采用 Weblogic 作为系统的应用服务器，或者支持 JDK1.4.2，TOMCAT4，SERVERLET, JDBC FOR MYSQL & MSSQL 虚拟主机；

● 客户端采用浏览器形式，满足零客户端的易用性要求；

● 资金允许条件下选用支持 PHP4&5 +Zend2.1,支持 DSO（APACHE 模块） 和 CGI 两种方式运行，这样在客户端维护，可以自由选择 PHP 脚本的运行方式。

2.4 需求概述

用户的主要权限：

注册；

登录；

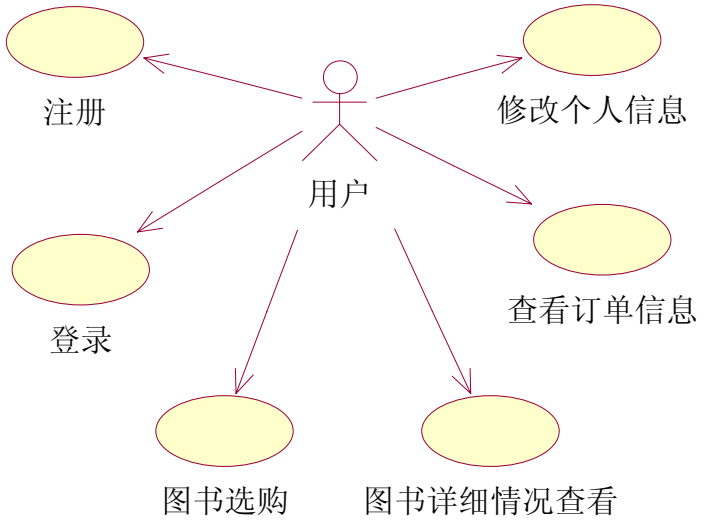
修改个人信息；

图书选购；

图书详细情况查看；

查看订单信息；

用户 UML 图



购物车主要功能：

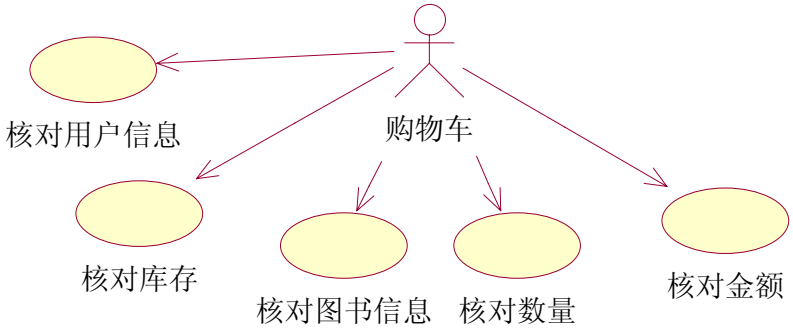
核对用户信息；

核对图书信息；

核对库存；

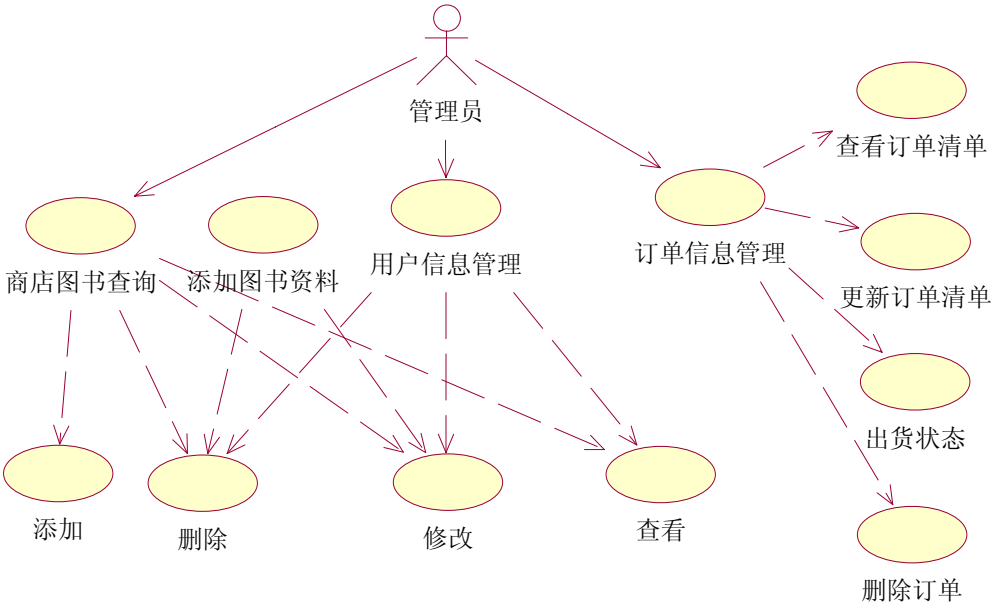
核对金额；

购物车 UML 图



管理员主要职能：

管理员 UML 图



3. 总体设计

3.1 总体结构设计

- 普通用户:

  用户注册;

  用户登录;

  查看用户的订单信息;

  修改用户个人信息;

  图书选购 (可按分类方式查找图书，或通过关键字进行查询);

  查看图书详细情况;

- 管理员:

  商店图书管理：添加，修改，删除，查看;

用户信息管理：查看，修改，删除;

订单信息管理：查看订单清单，更新订单付款，出货状态。删除订单。

- 购物车功能：核对库存，核对图书，核对数量，核对金额。

3.2 处理流程

普通用户：

首页(index.jsp)，浏览书目，注册，登录；

新用户注册,转到注册页面(reg.jsp)，要求填入个人信息，完成后回到首页(index.jsp)；

老用户输入用户名密码登录(login.jsp)，若错误转到错误页面(errorpge.jsp)；正确可选择更新个人资料(modimy.jsp)，进行购物；还可以查看已下的订单和货品状态(userinfo.jsp);

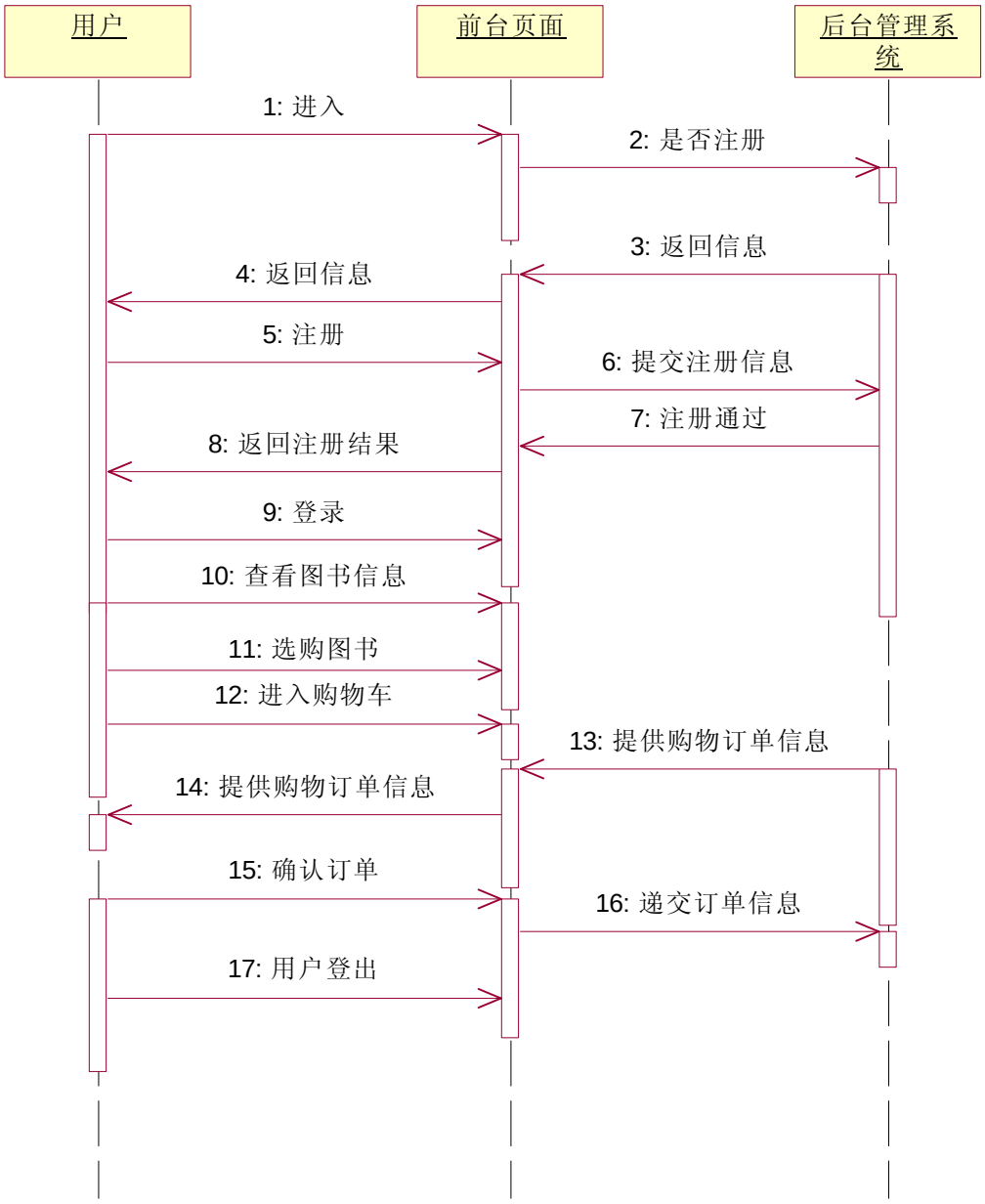
用户可在首页(index.jsp)分类中选择图书，转到图书列表页面(booklist,jsp)，详细介绍改图书；

确定购买该书，则转到购买页面(purchase.jsp)，要求输入数量，确定则转到最后确定页面(showbook.jsp)，否则回到首页(index.jsp)；

确定后转到购物车页面(shoperlist.jsp)，显示购物车物品清单；确

定货品和金额后，生成订单(showindent.jsp);

时序图



管理员：

通过首页(index.jsp)的网站管理进入后台管理页面(manage/..)；

进入管理员登录页面(manage/login.jsp)，要求输入用户名，密码；

正确进入后台管理主页面(manage/main.jsp)，错误则转到错误页面(manage/error.html);

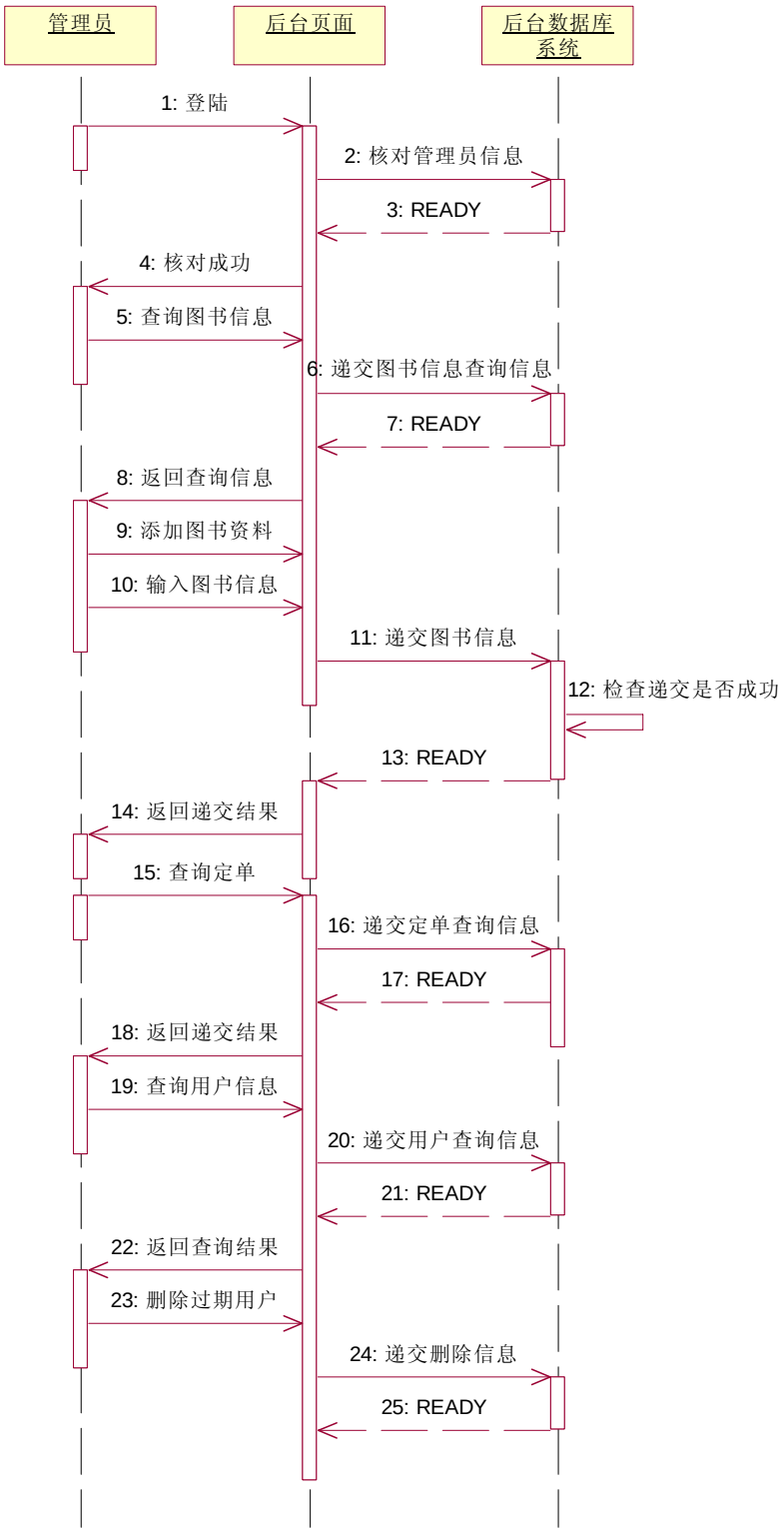
后台管理主页面(manage/main.jsp)，即时显示书店最新的订单情况，并可进行商店图书查询(manage/booklist.jsp)，添加图书资料(manage/addbook.jsp)，订单信息查询(manage/orderlist.jsp)，用户信息查询(manage/userlist.jsp)；完成后退出(loginout.jsp)；

商店图书查询(manage/booklist.jsp):显示书店当前图书情况，并可进行图书资料修改(manage/modibook.jsp)；点击图书名可以查看该书的作者，数量等属性(manage/showbook.jsp)；

订单信息查询(manage/orderlist.jsp):可以查看订单详细情况(manage/indentlist.jsp);

用户信息查询(manage/userlist.jsp):可以查看用户详细资料(manage/showuser.jsp),修改用户资料(manage/modiuser.jsp)。

时序图


管理员
后台页面
后台数据库系统
1: 登陆
2: 核对管理员信息
3: READY
4: 核对成功
5: 查询图书信息
6: 递交图书信息查询信息
7: READY
8: 返回查询信息
9: 添加图书资料
10: 输入图书信息
11: 递交图书信息
12: 检查递交是否成功
13: READY
14: 返回递交结果
15: 查询定单
16: 递交定单查询信息
17: READY
18: 返回递交结果
19: 查询用户信息
20: 递交用户查询信息
21: READY
22: 返回查询结果
23: 删除过期用户
24: 递交删除信息
25: READY

4. 数据库设计

4.1 数据库的分析

数据库的设计通常是以一个已经存在的数据库管理系统为基础的，常用的数据库管理系统有 Access、SQL Server、MySql、Oracle 等。根据用户需求和系统分析，本系统初步设计采用 SQL Server 2000 数据库管理系统。在 SQL Server 2000 数据库管理系统中建立名称为 bookdatabase 的数据库。

4.2 数据库的设计

4.2.1 数据库的设计结构

为满足系统需要，本系统总共建立了 My_Book、My_BookClass、My_BookAdminuser、My_Users、My_Indent 和 My_IndentList 6 个数据表。

表 5-1 数据库设计结构

表名	描述
My_Book	商店图书表
My_BookClass	商店图书类别表
My_BookAdminuser	商店管理员表
My_Users	购书用户表
My_Indent	购物车订单号表
My_IndentList	购物车订单货物列表

4.2.2 数据表的结构

4.2.2.1 商店图书表

商店图书表用来存储商店所有图书的信息，并设定 Id 为查询图书的主键值。

表 5-2 商店图书表

列名	数据类型	长度	允许空	说明
Id	int		NOT NULL	ID 序列号(主键值)
BookName	varchar	40	NOT NULL	书名
BookClass	int		NOT NULL	图书类别
Author	varchar	25	NULL	作者
Publish	varchar	150	NULL	出版社
BookNo	varchar	30	NULL	书号
Content	varchar	4000	NULL	内容简介
Price	float		NULL	书价
Amount	int		NULL	总数量
Leav_number	int		NULL	剩余数量
RegTime	datetime		NOT NULL	登记时间

4.2.2.2 商店图书类别表

此表主要用来分类图书，使图书可以根据不同的类别分开管理，提高效率。使用图书 Id 为主键值。

表 5-3 商店图书类别表

列名	数据类型	长度	允许空	说明
Id	int		NOT NULL	ID 序列号(主键值)
ClassName	varchar	30	NOT NULL	图书类别名

4.2.2.3 商店管理员表

此表是用来保存管理员用户名与密码的，当管理员登录时核对用户名与密码。

表 5-4 商店管理员表

列名	数据类型	长度	允许空	说明
AdminUser	varchar	20		管理员用户名(初始值 ‘admin’)
AdminPass	varchar	50		管理员密码(初始值 ‘nimda’)

4.2.2.4 购书用户表

此表是提供给用户，当其决定购买任一本书时，需要提交给网站的信息。

表 5-5 购书用户表

列名	数据类型	长度	允许空	说明
Id	int		NOTNULL	ID 序列号(主键值)
UserName	varchar	20	NOTNULL	购物用户名
PassWord	varchar	50	NOTNULL	用户密码
Names	varchar	20		用户联系用姓名
Sex	varchar	2		用户性别
Address	varchar	150		用户联系地址
Phone	varchar	25		用户联系电话
Post	varchar	8		用户联系邮编
Email	varchar	50		用户电子邮件
RegTime	DateTime			用户注册时间
RegIpAddress	varchar	20		用户注册时 IP 地址

4.2.2.5 购物车订单号表

此表为用户确定购买任一本书时，系统自动生成的订单。当用户填写了购书信息后，系统返回此信息。以便管理员以后进行核对。

表 5-6 购物车订单号表

列名	数据类型	长度	允许空	说明
Id	int		NOT NULL	ID 序列号(主键值)
IndentNo	varchar	20	NOT NULL	订单编号
UserId	int		NOT NULL	用户序列号
SumbitTime	datetime		NOT NULL	提交订单时间
ConsignmentTime	varchar	20		交货时间
TotalPrice	float			总金额
Content	varchar	400		用户备注
IpAddress	varchar	20		下单时 IP
IsPayoff	int		NULL	用户是否已付款
IsSales	int		NULL	是否已发货

4.2.2.6 购物车订单货物列表

此表提供购物车德订单号和购物车里的物品列表，方便管理员通知仓库发货。

表 5-7 购物车订单货物列表

列名	数据类型	长度	允许空	说明
Id	int		NOT NULL	ID 序列号(主键值)
IndentNo	int		NOT NULL	订单号表序列号
BookNo	int		NOT NULL	图书表序列号
Amount	int			订货数量

4.2.3 数据库的生成

完整的代码如下：

```
REM --SQL SERVER 网上书店数据库脚本文件  (订货单 indent)
/*
   ** Author:fiorano
   ** Create Time:2006.12.15

*/



drop table My_Book;

create table My_Book(         --商店图书表

   Id [INT] IDENTITY (1, 1) NOT NULL ,    --ID 序列号

   BookName varchar(40) NOT NULL,      --书名

   BookClass int not null,           --图书类别

   Author varchar(25) NULL,      --作者

   Publish varchar(150) NULL,    --出版社

   BookNo varchar(30) NULL,      --书号

   Content varchar(4000) NULL,      --内容介绍

   Price float NULL,       --书价

   Amount int NULL,        --总数量

   Leav_number int NULL,         --剩余数量

   RegTime datetime NOT NULl     --登记时间

   );

alter table My_Book           --设定 Id 为主键值
  add primary key (Id);
```

```
drop table My_BookClass;

create table My_BookClass(      --商店图书类别表
   Id [int] IDENTITY (1, 1) Not Null primary key,
--ID 序列号
   ClassName [varchar](30) not null          --图书类别
名
   );

drop table My_BookAdminuser;

create table My_BookAdminuser(     --商店管理员表
   AdminUser [varchar](20),      --管理员用户名
   AdminPass [varchar](50)       --管理员密码
   );
insert into My_BookAdminuser values('admin','nimda');
--初始化管理员用户

drop table My_Users;

create table My_Users (         --购书用户表
   Id [int] IDENTITY (1, 1) not null, --ID 序列号
   UserName varchar(20) not null,     --购物用户名
   [PassWord] varchar(50) not null,--用户密码
   Names varchar(20),         --用户联系用姓名
   Sex varchar(2),            --用户性别
   Address varchar(150),         --用户联系地址
```

```
    Phone varchar(25),          --用户联系电话
    Post varchar(8),          --用户联系邮编
    Email varchar(50),          --用户电子邮件
    RegTime  DateTime,          --用户注册时间
    RegIpAddress varchar(20)      --用户注册时 IP 地址
    );

alter table My_Users             --设定 Id 为主键值
  add primary key (Id);


drop table My_Indent;

create table My_Indent(             --购物车订单号表
    Id [int] IDENTITY (1, 1) not null, --ID 序列号
    IndentNo varchar(20) not null,      --订单编号
    UserId [int] not null,          --用户序列号
    SubmitTime datetime not null,    --提交订单时间
    ConsignmentTime varchar(20),     --交货时间
    TotalPrice float,         --总金额
    Content varchar(400),          --用户备注
    IpAddress varchar(20),          --下单时 IP
    IsPayoff [int] null,         --用户是否已付款
    IsSales [int] null          --是否已发货
    );
```

```
alter table My_Indent           --设定 Id 为主键值
  add primary key (Id);


drop table My_IndentList;

create table My_IndentList(        --购物车订单货物列表
   Id int IDENTITY (1, 1) not null,    --ID 序列号
   IndentNo [int] not null,         --订单号表序列号
   BookNo [int] not null,           --图书表序列号
   Amount [int]               --订货数量
);

alter table My_IndentList          --设定 Id 为主键值
  add primary key (Id);
----End The Create Table Sql
```